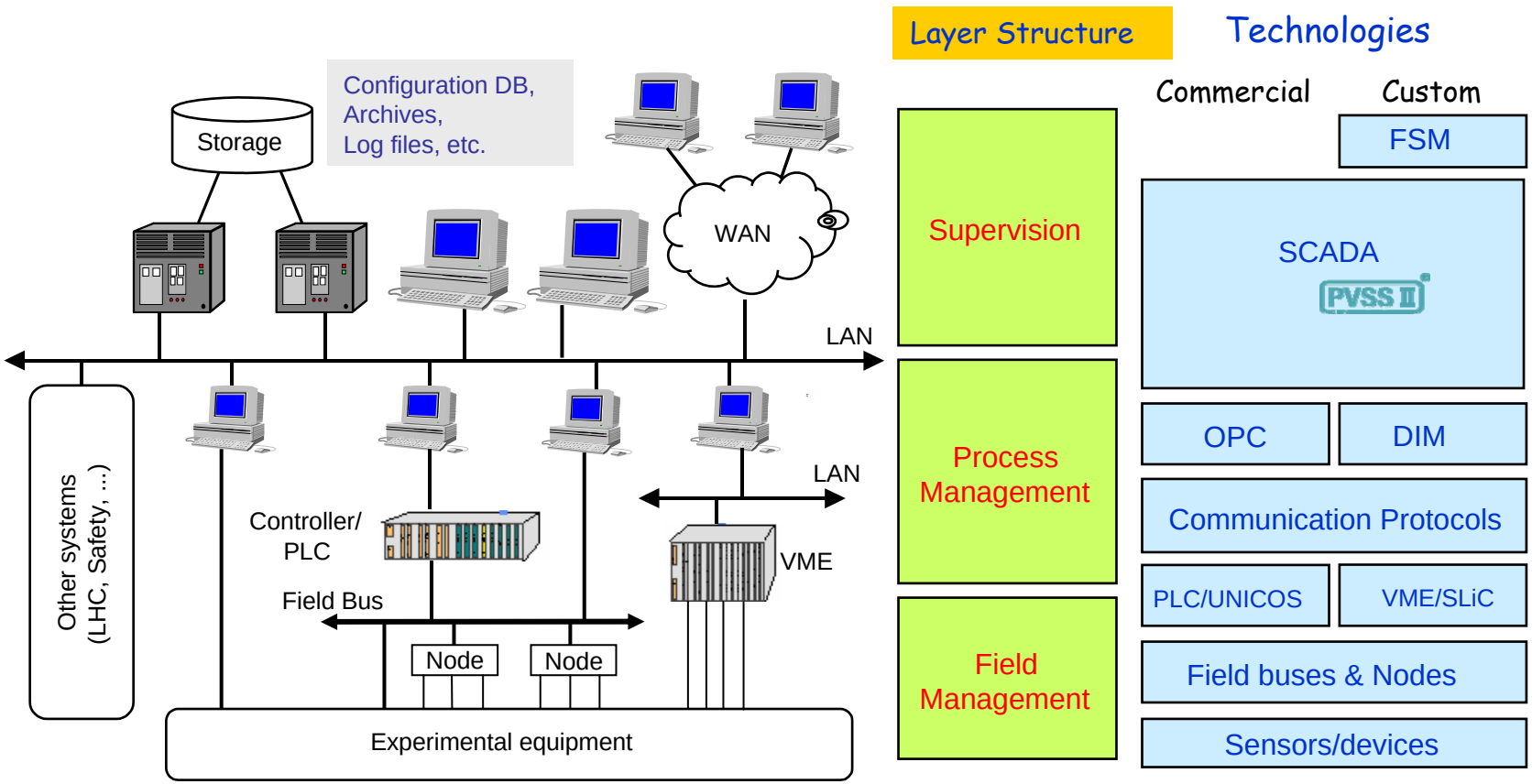
Detector Control Systems

A software implementation: Cern Framework + PVSS

Niccolo' Moggi and Stefano Zucchelli
University and INFN Bologna

Hardware and Software

Logically any DCS has a 3 layers hardware structure:
supervision -> control -> field



Requirements: architecture

- Client/Server architecture with hardware abstraction layers
  - Servers execute tasks and provide data independently of the clients
- Hierarchical mechanism (tree structure)
  - FSM (finite state machines): “nodes” with 1 parent and many children
  - ➢ easy partitioning
  - ➢ distributable system, possible decentralized decision making and error recovery

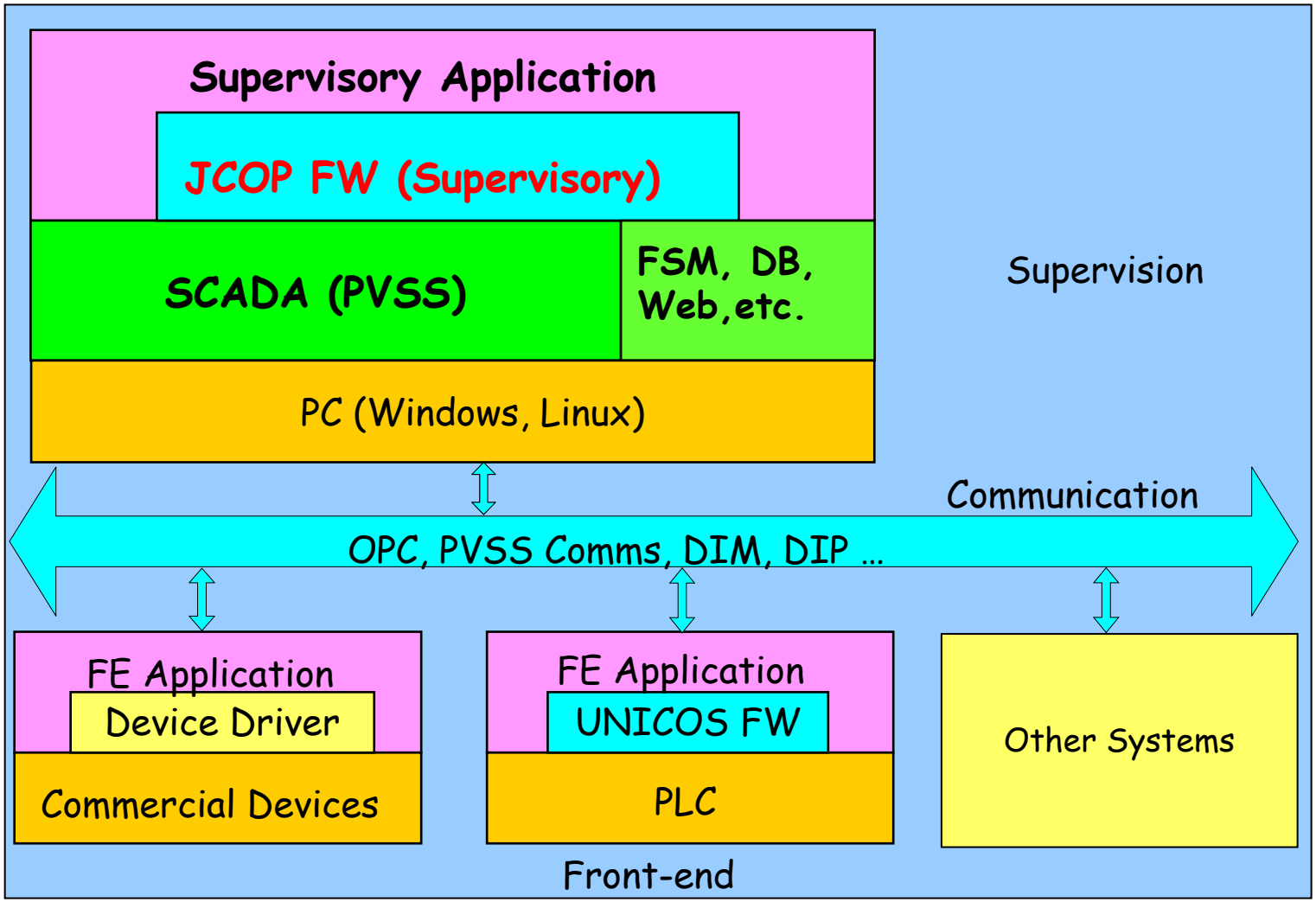
Requirements: implementation

- reliability
- flexibility, expandability
- low cost
- short development time
- ease of use (developers and users)
- documentation/support

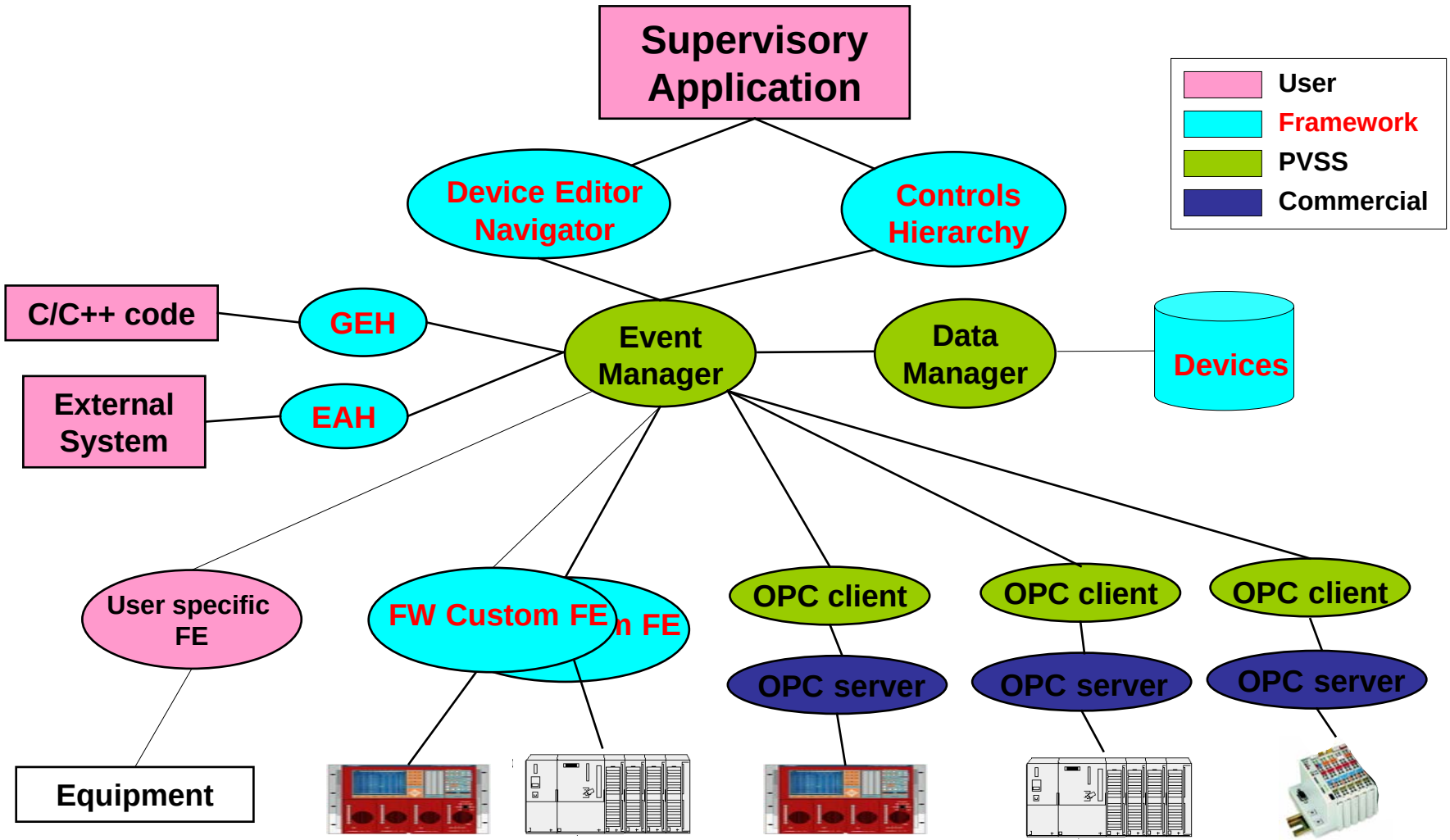
CERN choice: JCOP + PVSS

- What it is PVSS:
  - Commercial software by ETM (Austrian company)
  - SCADA (supervisory control and data acquisition)
    - Run-time DB, archiving, logging, trending
    - Alarm generation and handling
    - Device access (OPC/DIM, drivers) and control
    - User data processing (C-style scripting language)
    - Graphical editor for user interface
- What is JCOP (Joint COntrol Project)
  - CERN developed a framework for PVSS
    - Simple interface to PVSS
    - Implements hierarchy (FSM)
    - Provides drivers for most common HEP devices
    - Many utilities (eg: graphics)

JCOP + PVSS


Supervisory Application
JCOP FW (Supervisory)
SCADA (PVSS)
FSM, DB, Web,etc.
PC (Windows, Linux)
Supervision
Communication
OPC, PVSS Comms, DIM, DIP …
FE Application
Device Driver
Commercial Devices
FE Application
UNICOS FW
PLC
Other Systems
Front-end

JCOP + PVSS


Supervisory Application
User
Framework
PVSS
Commercial
Device Editor Navigator
Controls Hierarchy
C/C++ code
GEH
Event Manager
Data Manager
Devices
External System
EAH
User specific FE
FW Custom FE
OPC client
OPC client
OPC client
OPC server
OPC server
OPC server
Equipment

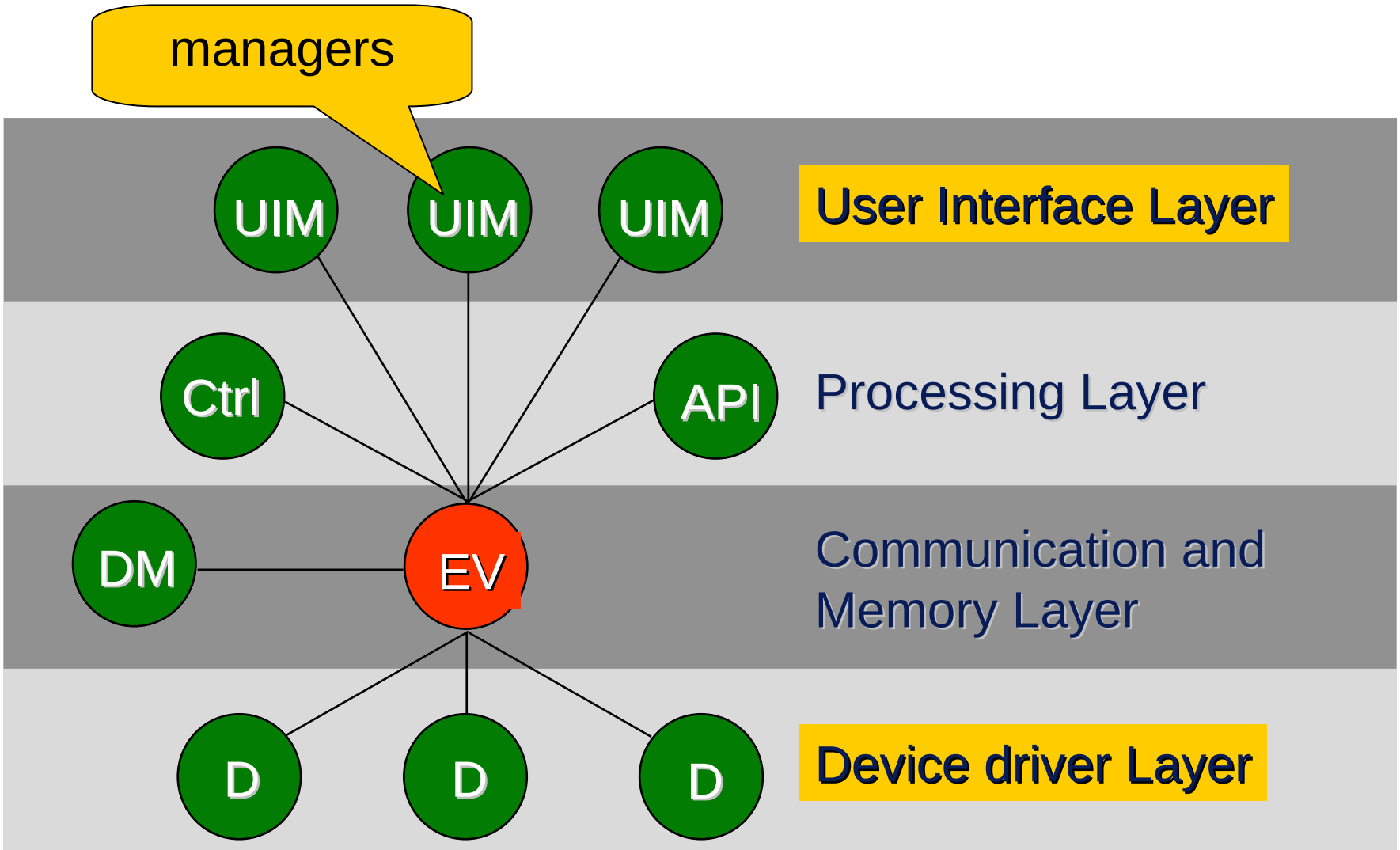
JCOP features

- Provides a complete component set for:
  - CAEN HV power supplies
  - Wiener crates and LV power supplies
  - Other power supplies (ISEG, ELMB)
  - Generic devices to connect analog or digital I/Os
- “complete” means:
  - any necessary OPC/DIM servers;
  - Device modeling (mapping of PVSS data-points to device values)
  - scripts, libraries and panels to configure and operate the device.
- Other tools to integrate user’s devices

PVSS II features

- Distributed architecture: several processes for different tasks (“managers”) run separately and communicate internally via TCP/IP
- Managers subscribe to data (“subscribe on change” mode)
- Event manager is the heart of the system
- Device oriented (abstraction of complex devices)
- Devices are modeled using “Data-Points”:
  - All data relative to a device is grouped together and may be reached hierarchically in C++ style (eg: crate.board.channel.vMon)

PVSS II system architecture



PVSS: how it was chosen

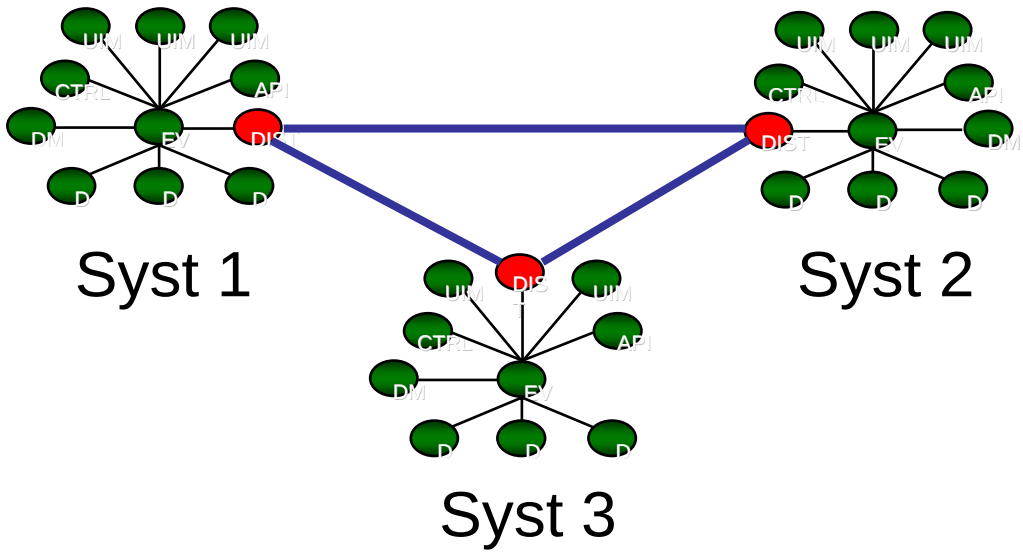
- CERN chose PVSS in 1999 among >40 products based on a set of “objective” criteria:
  - Scalability
  - Cross-platform
  - Archiving and trending ability
  - Remote access
  - Security
  - Alarm handling
  - Extensibility and ease of use
- It’s been a long evaluation process (lots of tests)
- Much development has been done since then

JCOP+PVSS advantages

- Scalability: practically no limits (see next page)
- Stability of kernel
- Flexibility (customization, easy integration of user functions)
- Win AND Linux (not OR)
    - managers of the same system may run on different platforms
    - may develop on one platform and run on the other
    - only limitation: OPC client/server must run on Win
- By now is tried and tested (Compass, Harp, NA60, all 4 LHC)
- HV and LV are “Plug and Play” (drivers and modeling)
- Easy partitioning (commissioning and calibrations)
- Documentation (cern site, not ETM)
- Easy remote access on the Web through a web server
  – user’s GUI get downloaded by the remote browser
  – claim to support any security option (well…)
- Very flexible alarm handling scheme
- DB: proprietary or Oracle
- Redundancy (double system with automatic switchover)
- Some nice safety features if system is overloaded

(follows) Scalability

- (Cern is) not aware of any built-in limit
  - As many managers as needed (all communications handled automatically)
  - Scattered system (one system running on many computers)
  - Distributed systems: multiple systems connected together and exchanging data



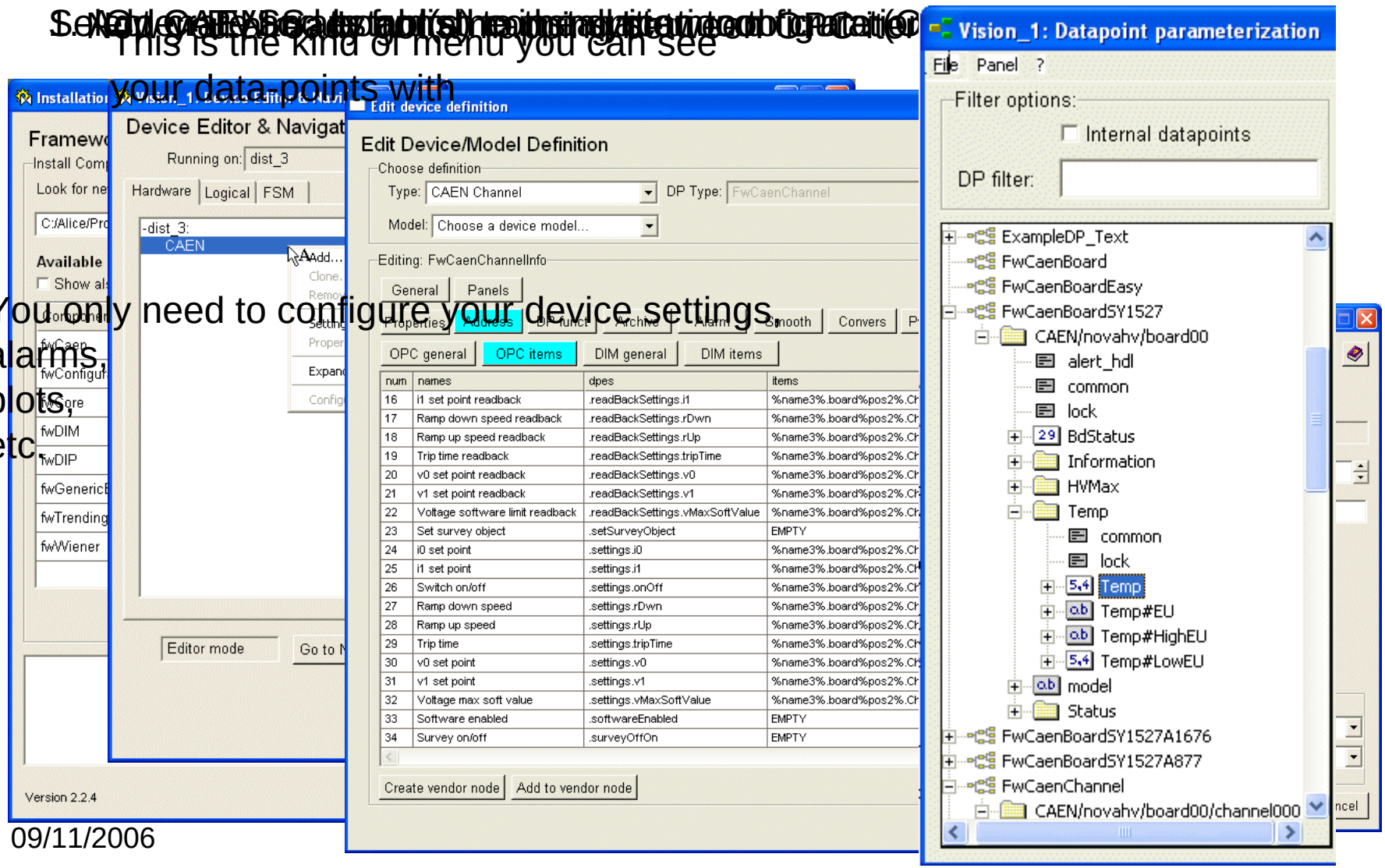
Dis-advantages

- COST !!!
  - have no idea: should investigate with ETM (koller@etm.at)
  - complex licensing model: usually possible to negotiate special deals
- Maybe even too “big” ?
- Still need to develop part of the device drivers (but this is unavoidable)
- Support by ETM in the US ?
- Cern will not commit to any *formal* support but `“this does not mean we will not help you if we could”` (Wayne.Salter@cern.ch)

Example: CAEN HV

This is the kind of menu you can see

your data-points with



You only need to configure your device settings, alarms, plots, etc

Performance

- Report tests made at Cern in 2003/6
  - Distributed 16-system in 3-level tree on 16 PC of various type, each system with ~40K DPE
  - Tested up to 260K DPE on one PC (non-realistic test 5M DPE on top PC of a 130-system in 5-level tree)
  - Total # of DPE not significant in subscribe-on-change mode. What overload a system is the # of changes/s
    - P4 CPU:2.4GHz RAM:2Gb running EM + DM + 5 managers
      - saturate CPU at 1600 changes/s (500 ch/s = 35%CPU)
      - when moving EM outside saturates CPU at 2800 changes/s
      - alarm handling and archiving: saturate at ~700 alarms/s
    - Dual-CPU and large RAM are well exploited
    - With 400K DPE (includes FEB)
      - 1% changes/s : 1+2 CPU tree
      - 5% changes/s : 1+10 CPU tree

Other possibilities

- EPICS: see Andrew Norman presentation
- iFix (by Intellution):
  - Commercial
  - Slow (see CDF experience)
  - Fragile connections between nodes
  - No drivers included
  - Windows only
  - Limit 100,000 channels ?
- LabView:
  - No large systems
- A lot of other commercial software out there, but should be tested…